Gesture Recognition: Case study IIITB & Upgrad

Problem Statement: We need to develop a cool feature in the smart-TV that can recognise five different gestures performed by the user which will help users control the TV without using a remote. The following table consists of the experiments done to build a model to predict the gestures from the given data set.

Exp .No .	Model	No. Of Trainable Parameters	Result/Accuracy	Comment
1.	Conv3D - Batch_size=64 - Activation function = 'elu' - Kernel_size=(3,3,3) - Using 84,84 image frames	9,439,365	categorical_accuracy: .7747 val_categorical_accuracy: 0.7222	Model little bit over-fitting. Tried with SGD ,lr = 0.001, Momentum = 0.9
2.	Conv3D - Batch_size=64 - Activation function = 'relu' - Kernel_size=(3,3,3) - Using last 18 image frames	17,803,397	categorical_accuracy: 0.1779 val_categorical_accuracy: 0.2083	Model is under-fitting. Changing the activation function did not improve accuracy,
3	Conv3D - Batch_size=64 - Activation function = 'relu' - Kernel_size=(3,3,3) - Using last 18 image frames		Negative Dimension Error.	The new CNN kernel sizes are not compatible with the output of previous layers. Let's reduce the kernel size of new layers.
4.	Conv3D - Batch_size=64 - Activation function = 'relu' - Kernel_size=(3,3,3) - Using last 18 image frames	9,360,389	categorical_accuracy: 0.1818 val_categorical_accuracy: 0.1944	Still there is no improvement in the model. after changing optimizer.
5.	Conv3D - Batch_size=64 - Activation function = 'elu' - Kernel_size=(3,3,3) - Using 18 frames	9,363,333	categorical_accuracy: 0.9526 val_categorical_accuracy: 0.7917	Model Over-fitting. Using batchnormalization improved model performance.

6.	Conv3D - Using (84X84) image frames - Activation - Relu	9,363,333	categorical_accuracy: 0.8696 val_categorical_accuracy: 0.7639	over-fitting using drop out 0.5
7.	Conv3D - Using (84,84) image frames -	42,522,501	categorical_accuracy: 0.9921 val_categorical_accuracy: 0.6944	Highly over-fitting Changing image size did not improve accuracy.
8.	Conv3D - Using (84X84) image frames - Learning Rate starting from 0.01 with Adam	710,533	categorical_accuracy: 0.6640 val_categorical_accuracy: 0.7361	Slight under-fitting.
9.	Conv2D TimeDistributed GRU -Using (84X84) images - Using 18 frames.	99,269	categorical_accuracy: 0.6601 val_categorical_accuracy: 0.6389	Under-fitting.With Drop Out 0.2 and Dense layer
10.	Conv2D + GRU - Using last 18 (84X84) images per video - Using Adam optimizer	99,269	categorical_accuracy:0.6443 val_categorical_accuracy: 0.6389	No Over or Under fitting but Less Accuracy
11.	Conv2D + GRU - Adding more layers - Using last 18 (84X84) images per video	128,517	categorical_accuracy: 0.8458 val_categorical_accuracy: 0.7222	Under-fitting. With Gloabavaerage3D and Dense layer
12.	- Timedistributed ConvLSTM	13,589	categorical_accuracy: 0.7866 val_categorical_accuracy: 0.8472	This is the best model so far we can get. The validation accuracy is good and the numbers of parameters are 13,589.
13.	Conv3D - Using alternative 18 (84X84) images - Updating Epochs to 50	9,439,365	categorical_accuracy: 0.7391 val_categorical_accuracy: 0.6667	Slightly over-fitting.

Best Model: - TimeDitributed ConvLSTM

We have selected model from experiment 12 as our final model for the following reasons.

Using mean subtraction as normalizing technique for the batch gave substantially better performance than dividing pixels by 255. Model is able to capture the gesture from the alternative frames than last 18 consecutive frames.

We used epoch=50 and batch size=64 for limitation of computational resources.at 47 No epochs given highest accuracy with 0.86 both.

Adding further dropouts is not improving performance.So tried with TimeDistributed ConvLSTM.

Most importantly both training and validation accuracy > 0.75 and very low difference between the 2, signifying that there is under-fitting .